



Les ontologies pour la recherche ciblée d'information sur le Web : une utilisation et extension d'OWL pour l'expansion de requêtes

Nicolas Guelfi, Cédric Pruski, Chantal Reynaud

► To cite this version:

Nicolas Guelfi, Cédric Pruski, Chantal Reynaud. Les ontologies pour la recherche ciblée d'information sur le Web : une utilisation et extension d'OWL pour l'expansion de requêtes. Ingénierie des connaissances 07 (IC07), Jul 2007, Grenoble, France. pp.61-72. hal-00382537

HAL Id: hal-00382537

<https://hal.science/hal-00382537>

Submitted on 9 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Les ontologies pour la recherche ciblée d'information sur le Web : une utilisation et extension d'OWL pour l'expansion de requêtes

Nicolas Guelfi¹, Cédric Pruski^{1,2,3} et Chantal Reynaud^{2,3}

¹Laboratory for Advanced Software Systems, Université du Luxembourg,
{nicolas.guelfi, cedric.pruski}@uni.lu

²LRI ; Univ. Paris-Sud, CNRS ; Bât 490, 91405 Orsay, France

³INRIA Futurs; projet Gemo, Parc Club Orsay Université, 4 rue Jacques Monod, 91894 Orsay
{cedric.pruski, chantal.reynaud}@lri.fr

Résumé: L'utilisation des technologies du Web Sémantique s'est généralisée au cours de ces dernières années. Ceci est vrai, en particulier, pour les langages de définition d'ontologies nécessitant l'adaptation des outils basés sur l'exploitation de ce type de ressources terminologiques pour qu'ils fonctionnent à partir de connaissances représentées dans le langage standard OWL. Cet article concerne la recherche d'information sur le Web basée sur l'utilisation d'ontologies. Notre contribution est double. Tout d'abord nous étudions comment exploiter les connaissances représentées en OWL dans O³, une approche de recherche de documents sur le Web, et nous proposons d'étendre ce langage en intégrant de nouvelles relations sémantiques. Ensuite, nous montrons comment utiliser les ontologies construites en OWL et son extension pour l'expansion de requêtes afin d'optimiser la précision des résultats lors de recherche d'informations sur le Web. Quelques résultats d'expérimentations obtenus grâce au prototype TARGET sont présentés.

Mots-clés: Ingénierie des connaissances, Ontologie, Web Sémantique, Langage de requêtes.

1 Introduction

Depuis l'avènement de l'Internet dans les années 90, le nombre toujours croissant de documents formant « la toile » nécessite l'utilisation d'outils adaptés afin d'assister les utilisateurs dans leur tâche de recherche d'information pertinente. Cet objectif fait également partie de la vision du Web Sémantique (Berners-Lee et al., 2001). Ce dernier, par l'intermédiaire de concepts tels que les ontologies, essaye de donner un sens aux données du Web ce qui, si cet objectif se réalise, facilitera de manière considérable la recherche de documents pertinents.

Dans nos précédents travaux (Guelfi & Pruski, 2006), nous avons proposé l'approche générale O³ basée sur WordNet afin d'optimiser, en terme de pertinence des résultats, la recherche de documents sur le Web. Elle est, entre autres, basée sur l'ajout de termes dans les requêtes, ces termes étant extraits du vocabulaire du

domaine de recherche modélisé dans WordNet (Fellbaum, 1998). O^3 a été développée dans un cadre formel basé en partie sur la logique et la théorie des graphes. Cette formalisation a permis une définition rigoureuse et non ambiguë des règles d'affinement de requêtes (Guelfi & Pruski, 2006). Par ailleurs, l'adoption du langage OWL (McGuinness & Van Harmelen, 2004) par le W3C en tant que recommandation depuis 2004, a provoqué le développement rapide et massif d'ontologies dans ce format. Afin de bénéficier de ce phénomène et de l'approche O^3 prometteuse, nous avons choisi de rendre cette dernière compatible avec l'utilisation d'ontologies OWL. Ce travail consiste à intégrer dans O^3 l'exploitation d'ontologies, des relations d'équivalence, de subsomption, d'instanciation de OWL et à proposer d'intégrer l'exploitation de nouvelles primitives, les relations de composition¹ et d'opposition, celles-ci n'étant pas définies dans le langage selon la recommandation du W3C. Notre but est que l'approche O^3 puisse atteindre les mêmes objectifs que dans sa version initiale mais puisse également être améliorée grâce à l'expressivité de OWL. Néanmoins, notre volonté n'est pas uniquement d'exploiter toute la puissance d'expression du langage mais également son utilité et son efficacité pour nos travaux futurs portant sur l'évolution et l'adaptation des ontologies décrits dans (Guelfi et al., 2007).

Nous proposons, dans cet article, d'une part une étude des possibilités de représentation des connaissances dans OWL que nous proposons d'intégrer dans O^3 ainsi qu'une extension de ce langage, d'autre part, une exploitation de OWL étendu pour l'expansion de requêtes au sein du système prototype TARGET. Ainsi, la section suivante introduit l'approche O^3 et présente notre proposition d'extension du langage OWL. La troisième partie discute de l'approche d'expansion de requêtes pour le Web basée sur l'utilisation des ontologies OWL. Nous présentons ensuite quelques résultats expérimentaux. L'article s'achève sur une conclusion et l'énoncé des perspectives de ces travaux.

2 L'utilisation du standard OWL et son extension

L'approche O^3 autorise l'utilisation des relations sémantiques suivantes : la synonymie, la méronymie, l'hyperonymie et l'antonymie. Ces relations qui sont celles représentées dans WordNet correspondent à des relations entre termes pour la synonymie ou entre concepts (synsets dans WordNet) pour les autres relations. Notre travail a donc consisté, dans un premier temps, à étudier la représentation de ces différentes relations dans OWL. La synonymie, étant une relation entre termes, n'est pas représentée. En revanche, nous exploiterons la relation d'équivalence entre concepts représentée dans OWL. La relation d'hyperonymie trouve son correspondant dans la relation de subsomption. En revanche, les deux relations restantes, la méronymie et l'antonymie, ne font pas partie des primitives du langage OWL dans

¹ Le terme « composition » doit être pris ici au sens général du terme. Il ne présuppose aucune interprétation.

l'état actuel de la recommandation du W3C. Dans ce paragraphe, nous présentons tout d'abord l'approche O^3 . La section 2.2 portera sur les primitives d'OWL utilisées dans la nouvelle version² de O^3 , appelée O^4 , et l'extension proposée du langage sera présentée en section 2.3.

2.1 L'approche O^3

L'approche O^3 (Guelfi & Pruski, 2006), schématisée figure 1, a pour but d'améliorer, en termes de pertinence, les résultats retournés lors de recherche d'informations sur le Web. Un premier prototype de l'approche a été construit et a montré des résultats prometteurs (voir section 4). L'outil utilise Google comme interface avec le Web et le langage de requêtes ASK que nous avons développé (Guelfi & Pruski, 2006). Une première requête non enrichie est soumise à Google afin d'extraire une première série de pages Web. L'ensemble des pages ainsi obtenu est converti en WPGraph et W^3 Graph, des structures formelles du Web construites selon une ontologie. Une deuxième requête, enrichie cette fois, par application de règles d'expansion de requêtes basées sur la même ontologie, est vérifiée sur ces graphes afin de sélectionner les pages les plus pertinentes. Les pages Web renvoyées correspondent davantage aux attentes de l'utilisateur que celles retournées par les moteurs de recherche usuels en vertu des règles d'expansion qui ont pour effet, grâce aux termes ajoutés à la requête initiale, de cibler davantage l'espace de recherche.

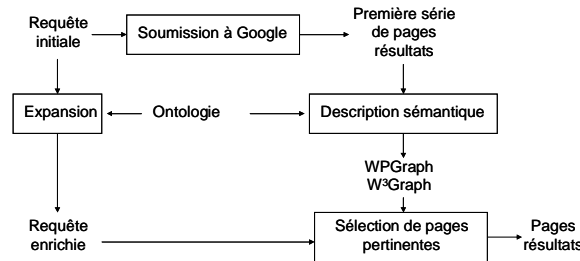


Fig. 1 – L'approche O^3

Un WPGraph est un graphe non orienté, non connexe où les sommets représentent les concepts d'une page Web et où les arêtes représentent les liens sémantiques entre les concepts. Les arêtes sont construites suivant une ontologie et sont pondérées suivant la métrique de Hirst-St-Onge (Hirst & St-Onge, 1998). Intuitivement, il y a une arête entre 2 sommets du graphe si le chemin minimal entre les deux concepts dans l'ontologie n'excède pas un certain poids donné par application de la métrique. De même, un W^3 Graph est un graphe dont les sommets sont des WPGraphs et les arêtes représentent un lien sémantique entre les sommets. Si deux sommets ont un contenu très proche du point de vue sémantique, ils seront reliés avec un poids important. Plus formellement, un WPGraph est un sextuplet $(V, E, T, \varphi, \rho_v, \rho_e)$ où V

² O^3 : Optimal Ontology-based Web IR

O^4 : Optimal OWL Ontology-based Web IR

représente l'ensemble des concepts d'une page Web, E l'ensemble des arêtes (ensemble construit suivant une ontologie), T un ensemble de types (image, vidéo, etc.), ϕ une fonction d'étiquetage des sommets, ρ_v et ρ_e des fonctions de pondération des sommets et des arêtes basées sur la métrique de Hirst-St-Onge. Un W^3 Graph est un triplet (S, A, ρ) où S représente un ensemble de sommets (les WPGraphs des pages Web correspondantes), A un ensemble d'arêtes et ρ une fonction de pondération des arêtes basée également sur la métrique de Hirst-St-Onge.

2.2 Les primitives d'OWL utilisées dans O⁴

Les relations de base présentes dans OWL, que sont l'équivalence et la subsumption, constituent l'essence même de bon nombre de langage de représentation des connaissances. La recommandation du W3C sur OWL propose plusieurs primitives afin d'établir ces relations, soit entre concepts soit entre instances (voir table 1).

Le langage permet également de définir des instances de concepts préalablement définis (relation d'instanciation) et offre la possibilité de raisonner dessus grâce notamment aux primitives de la table 1. L'instanciation est intéressante dans un objectif d'expansion de requêtes Web car elle permet de préciser davantage la requête. Supposons, pour illustrer l'utilité de cette notion d'instanciation, qu'un utilisateur souhaite obtenir des informations sur le poids d'une Guilia (voiture de la marque Alpha Romeo). Si une requête ne contient que *poids Guilia*, beaucoup de pages retournées concerneront des personnes dont le prénom est Guilia. En revanche, si une ontologie du domaine automobile contient Guilia en tant qu'instance de voiture, et qu'il a été explicitement déclaré que l'interprétation de cette requête devait être faite dans le domaine de l'automobile, l'ajout du concept voiture servira de filtre et permettra de ne retenir que les pages concernant les voitures Guilia. Pour ces raisons, la relation d'instanciation, qui initialement n'était pas intégrée dans l'approche O³, a été rajoutée ainsi que la possibilité, dans le langage de requête ASK, de spécifier le domaine d'application auquel se réfère la requête (cf. section 3.1).

Dans notre approche, nous souhaitons pouvoir construire une ontologie au sein de laquelle il est possible de modéliser le contenu d'une requête correspondant à une expression composée d'un adjectif associé à un nom (i.e. le bras rouge). OWL distingue la représentation des propriétés qui relient des individus à des valeurs de données (Datatype property) des propriétés qui relient des individus à d'autres individus (Object Property). Si une propriété n'a pas de caractère d'abstraction (i.e. nous ne souhaitons pas considérer la notion de couleur ou bien ne souhaitons pas avoir plusieurs instances de rouge), elle ne correspondra pas dans l'ontologie à une relation entre individus. Il s'agit d'un attribut, représenté en OWL par une propriété de type Datatype Property. De façon générale, les valeurs d'attributs sont très discriminantes sur un domaine donné. Elles sont donc très utiles dans notre approche pour réduire l'ensemble des résultats d'une recherche (i.e. il y a moins de pages parlant de « bras rouge » que de pages parlant de « bras »).

Table 1. Relations de base dans OWL

Relation		Syntaxe abstraite	OWL	DL
Equivalence	Classe	EquivalentClasses($C_1 \dots C_n$)	owl:equivalentClass owl:sameAs	$C_1 = \dots = C_n$
	Instance	SameAs($I_1 \dots I_n$)	owl:sameAs owl:sameIndividual	$I_1 = \dots = I_n$
Subsumption	Classe	SubClassOf(C_1, C_2)	rdfs:subClassOf	$C_1 \sqsubseteq C_2$

Dans le tableau ci-dessus, les primitives OWL sont celles données dans la recommandation du W3C. Les C_i représentent des classes et les I_i des instances.

2.3 Proposition d'extension d'OWL

La relation de méronymie est une relation hiérarchique partitive. En d'autres termes, c'est la relation qui relie une partie à un tout, comme par exemple une voiture (tout) et une roue (partie). Cette relation est exploitée dans l'approche O^3 mais cette relation n'existe pas dans OWL. La méronymie correspond à une relation relativement générale. En effet, les liens entre composant/objet, membre/collection ou encore matériau/objet sont tous qualifiés de liens de méronymie alors que la relation de composition liant le composant au composé n'est pas exactement identique. Il convient donc de considérer plusieurs types de relations de méronymie comme le fait, par exemple, UML (OMG, 2005), en distinguant l'agrégation de la composition. Ces deux types de relation se différencient par le fait qu'une instance d'une partie d'un objet composite ne peut appartenir à un autre objet (agrégat ou composite).

Une manière d'exprimer la méronymie dans le langage OWL consiste soit à utiliser des collections d'objets (via les primitives containers en RDF) ou la réunion de plusieurs concepts (avec la primitive owl:unionOf). Ces solutions permettent de représenter la notion d'agrégat au niveau des classes (avec owl:unionOf) et au niveau des instances (avec les containers). Cependant, ces solutions sont très peu intuitives et accroissent la complexité du raisonnement. Par ailleurs, il n'est pas possible de définir la composition de cette manière car une instance impliquée dans les relations mises en œuvre par les containers ou par la primitive owl:unionOf n'est pas obligatoirement reliée uniquement au concept agrégat, d'où la nécessité d'introduire une primitive spécifique dans notre approche. La sémantique que nous adoptons pour cette primitive correspond à celle proposée dans UML (Barbier et al., 2003) pour l'agrégation et la composition (cf. ci-dessus). La composition et l'agrégation sont toutes deux transitives et asymétriques. Les contraintes de cardinalité sont en revanche différentes, d'où l'extension pour OWL proposée table 2 où les C_i sont des classes OWL, x , y et z des instances de classe. On utilisera $C_i(x)$ dans les relations (i.e. $\text{Agg}(C_1(x), C_2(y))$) pour faciliter la lisibilité³.

³ Il faudrait écrire $\text{Agg}(C_1, x, C_2, y)$ et l'axiome $\text{Agg}(C_1, x, C_2, y) \rightarrow C_1(x) \wedge C_2(y)$. Ceci vaut également pour composedOf.

Table 2. Balises OWL pour l'agrégation et la composition

Balise	Syntaxe abstraite	Logique du premier ordre
tg:Agg	Agg(C C')	1) $\forall C_1, C_2, C_3 \forall x, y, z \text{ Agg}(C_1(x), C_2(y)) \wedge \text{Agg}(C_2(y), C_3(z)) \rightarrow \text{Agg}(C_1(x), C_3(z))$ 2) $\forall C_1, C_2 \forall x, y \text{ Agg}(C_1(x), C_2(y)) \rightarrow \neg \text{Agg}(C_2(y), C_1(x))$ 3) $\forall C_1, \exists x, y, 1 \leq \{C_2(y) \mid \text{Agg}(C_1(x), C_2(y))\} $ 4) $\forall C_2, \exists x, y, 1 \leq \{C_1(x) \mid \text{Agg}(C_1(x), C_2(y))\} $
tg:composedOf	composedOf(C C')	1) $\forall C_1, C_2, C_3 \forall x, y, z, \text{ composedOf}(C_1(x), C_2(y)) \wedge \text{composedOf}(C_2(y), C_3(z)) \rightarrow \text{composedOf}(C_1(x), C_3(z))$ 2) $\forall C_1, C_2 \forall x, y, \text{ composedOf}(C_1(x), C_2(y)) \rightarrow \neg \text{composedOf}(C_2(y), C_1(x))$ 3) $\forall C_2, \forall x, y, \{C_1(x) \mid \text{composedOf}(C_1(x), C_2(y))\} = 1$ 4) $\forall C_1, \forall x, y, 1 \leq \{C_2(y) \mid \text{composedOf}(C_1(x), C_2(y))\} $

L'autre relation que nous proposons d'intégrer à OWL est l'opposition. S'apparentant à l'antonymie, cette relation permet de spécifier l'antagonisme entre concepts, relations, attributs et instances. C'est une relation complexe qui revêt différentes formes. La première est dite « opposition complémentaire » (pair/impair, présence/absence, etc.). Selon cette forme d'opposition, l'affirmation de l'un des termes entraîne nécessairement la négation de l'autre. La seconde, opposition entre des termes dits « mesurables » (petit/grand, chaud/froid, etc.), est fortement dépendante du contexte et de la valeur de référence des attributs qualifiés. Ce type d'opposition s'applique principalement sur des propriétés. Une troisième forme d'opposition affecte les valeurs spatio-temporelles et culturelles que l'on attribue aux termes (soleil/lune, départ/arrivée). Ce dernier type d'antonymie est important pour caractériser l'opposition entre concepts comme, par exemple, le bruit et le silence mais aussi entre relations comme départ et arrivée. Enfin, dans certains contextes, l'opposition concerne des instances, par exemple le fait d'opposer des personnages particuliers comme Laurel et Hardy.

La notion d'opposé est très utilisée en recherche d'information dans la vie courante et qui plus est, peu d'applications informatiques la mettent en œuvre. Ceci est particulièrement vrai pour les principaux moteurs de recherche Yahoo ou Google. Ces derniers permettent d'exprimer directement la négation mais pas l'opposition entre les termes d'une requête. Ainsi, si un utilisateur est intéressé pour avoir des informations sur la cuisine sucrée, par exemple, et que salée est l'antonyme de sucrée, le système devra être capable d'interpréter que l'utilisateur ne cherche pas de documents sur la cuisine salée. Afin de caractériser cette relation, nous nous appuyons sur les axiomes proposés par le linguiste Edmundson (Edmundson, 1967). Ce dernier définit l'antonymie comme une relation irréflexive, symétrique, antitransitive, identité droite et non-vide. Cette dernière propriété forcerait tous les concepts d'une ontologie à avoir un antonyme, nous avons décidé de ne pas prendre en compte cette propriété dans notre définition de l'antonymie. La table 3 donne la sémantique en logique du premier ordre de l'opposition en OWL que nous proposons. Dans cette table, les C_i et

les I_i sont des prédicats unaires et les R_i sont des relations. `contraryOf` est un prédicat binaire, `sameAs`, `equivalentClass` et `equivalentProperty` sont des primitives OWL.

Table 3. Opposition en OWL

Balise	Syntaxe abstraite	Logique du premier ordre
tg:contraryOf	<code>contraryOf(C₁ C₂)</code>	1) $\forall C \neg \text{contraryOf}(C, C)$ 2) $\forall C_1, C_2 \text{ contraryOf}(C_1, C_2) \rightarrow \text{contraryOf}(C_2, C_1)$ 3) $\forall C_1, C_2, C_3 \text{ contraryOf}(C_1, C_2) \wedge \text{contraryOf}(C_2, C_3) \rightarrow \text{equivalentClass}(C_1, C_3)$ 4) $\forall C_1, C_2, C_3, \text{ contraryOf}(C_1, C_2) \wedge \text{equivalentClass}(C_2, C_3) \rightarrow \text{contraryOf}(C_1, C_3)$
	<code>contraryOf(I₁ I₂)</code>	1) $\forall I \neg \text{contraryOf}(I, I)$ 2) $\forall I_1, I_2 \text{ contraryOf}(I_1, I_2) \rightarrow \text{contraryOf}(I_2, I_1)$ 3) $\forall I_1, I_2, I_3 \text{ contraryOf}(I_1, I_2) \wedge \text{contraryOf}(I_2, I_3) \rightarrow \text{sameAs}(I_1, I_3)$ 4) $\forall I_1, I_2, I_3, \text{ contraryOf}(I_1, I_2) \wedge \text{sameAs}(I_2, I_3) \rightarrow \text{contraryOf}(I_1, I_3)$
	<code>contraryOf(R₁ R₂)</code>	1) $\forall R \neg \text{contraryOf}(R, R)$ 2) $\forall R_1, R_2 \text{ contraryOf}(R_1, R_2) \rightarrow \text{contraryOf}(R_2, R_1)$ 3) $\forall R_1, R_2, R_3 \text{ contraryOf}(R_1, R_2) \wedge \text{contraryOf}(R_2, R_3) \rightarrow \text{equivalentProperty}(R_1, R_3)$ 4) $\forall R_1, R_2, R_3, \text{ contraryOf}(R_1, R_2) \wedge \text{equivalentProperty}(R_2, R_3) \rightarrow \text{contraryOf}(R_1, R_3)$

3 L'expansion de requêtes

L'idée d'affinement de requêtes n'est pas nouvelle. Plusieurs approches utilisent différentes techniques pour sélectionner les termes à ajouter à une requête. La plus répandue d'entre elles est basée sur des analyses statistiques de corpus de documents (Cui et al., 2002). L'objectif est de relever la fréquence des termes apparaissant conjointement sur un même document et de sélectionner les termes avec le plus grand coefficient. Une autre approche consiste à tenir compte du retour de l'utilisateur. Plus simplement, le système propose, suite à une requête, un ensemble de documents et suivant ceux visualisés par l'utilisateur, le système met à jour son index de termes concordant par des méthodes d'apprentissage automatique (Lin, 2006). La dernière approche figurant dans la littérature met en œuvre des ressources terminologiques telles que des ontologies ou des thésaurus contenant le vocabulaire servant à l'enrichissement des requêtes. Cette solution a le mérite d'être utilisable directement sans phase d'analyse ou d'apprentissage. Les approches de ce type existantes utilisent plutôt des ontologies mais uniquement les relations d'équivalence et de subsumption (Navigli & Velardi, 2003) ou encore de méronymie pour les plus récentes d'entre elles (Necib & Freytag, 2004) mais elles ne tirent pas parti de toutes les relations sémantiques pouvant exister entre tous les éléments d'une ontologie. Notre approche

est novatrice par la nature et la diversité des relations sémantiques exploitées (équivalence, subsumption, instance, composition et opposition).

3.1 Le langage ASK

Les requêtes que nous proposons d'enrichir sont construites dans le langage ASK (Guelfi & Pruski, 2006) dont la sémantique, donnée en logique du premier ordre, respecte le formalisme des WPGraphs et W³Graphs car les requêtes ASK seront ensuite vérifiées sur la structure logique de ces graphes. Nous présentons ici l'adaptation de ASK compte tenu des nouvelles primitives introduites en section 2 pour représenter l'ontologie afin, d'une part que l'utilisateur puisse bénéficier de plus d'opérateurs pour construire ses requêtes et aussi d'un point de vue expansion de requêtes afin que les termes ajoutés précisent davantage la requête. Les modifications par rapport à la version précédente au sein du langage ASK concernent : (1) l'intégration de l'opérateur « – » pour exprimer la notion d'antonymie entre les termes de la requête (ex : cuisine– sucré pour spécifier le contraire de la notion de sucré en cuisine), (2) l'intégration des attributs, par exemple, pour rechercher une voiture de couleur rouge, on écrira voiture.couleur(=rouge), (3) l'ajout de l'opérateur « : » qui, suivi d'une chaîne de caractères, sert à préciser le domaine dans lequel est soumise la requête (ex : roue:automobile pour rechercher des informations sur les roues dans le domaine de l'automobile). Cette dernière fonctionnalité n'est pas supportée actuellement par les langages des moteurs de recherche couramment utilisés.

3.2 Règles d'expansion de requêtes

L'idée derrière les techniques d'expansion de requête est d'assister l'utilisateur à construire de bonnes requêtes, c'est-à-dire des requêtes qui lui donneront les résultats escomptés. Dans notre approche, nous sélectionnons le vocabulaire proposé par une ontologie modélisant le domaine de recherche, afin de l'ajouter aux termes initiaux de la requête. Les règles que nous proposons sont fondées tout d'abord sur les relations ontologiques présentées en section 2 puis sur l'exploitation des propriétés des connecteurs logiques de ASK. L'étude menée par Joho (Joho et al., 2002) portant sur la pertinence des résultats d'une recherche en fonction de la façon dont la requête a été étendue nous permet de privilégier certaines formes d'expansion au dépend d'autres. L'étude montre que les relations de synonymies entre termes associées à un même concept sont privilégiées par les utilisateurs, viennent ensuite dans l'ordre les relations de subsumption entre concepts, les relations partie-de et d'opposition. C'est pourquoi, nous avons établi des priorités entre les relations ontologiques considérées pour étendre les requêtes. L'ordre selon lequel celles-ci sont considérées est le suivant : (1) relations d'équivalence, (2) subsumption (ajout des concepts plus généraux) et instanciation, (3) composition (composition stricte et agrégation), (4) opposition. Etant donnée une requête, les ajouts ne portent que sur les concepts liés à ceux qui sont déjà présents par une seule de ces relations, celle présente dans l'ontologie ayant la plus forte priorité. Enfin, le nom du domaine de recherche est également introduit dans la requête. Ce dernier a un fort caractère discriminant car le fait de forcer le

système à trouver des pages contenant en plus des termes de la requête le nom du domaine permet de bien filtrer les résultats. Concernant la relation d'instanciation (non traitée par l'étude de Joho), nous avons considéré qu'elle avait la même priorité qu'une relation de subsumption. En effet, dans notre exemple précédent, une Alpha Romeo Guilia peut être considérée comme une instance du concept voiture mais aussi comme une sorte de voiture (classe liée à voiture par une relation de subsumption). Ainsi, dans notre implémentation, le choix entre subsumption et instanciation est fait aléatoirement. Ensuite, les connecteurs logiques de ASK et en particulier la conjonction, sont exploités de façon à ce que les termes ajoutés à la requête aient pour effet de contraindre davantage l'espace des résultats, ce qui donnera à l'utilisateur des informations plus précises par rapport à la requête posée initialement. Un gain de temps sera obtenu, l'utilisateur n'ayant plus à filtrer manuellement les résultats, comme c'est le cas avec les moteurs de recherche usuels.

Illustrons le fonctionnement de nos règles sur l'exemple de la requête ASK *publication&tree:computer*⁴ (pour obtenir des pages sur les publications sur les arbres au sens de la théorie des graphes) en supposant que, dans notre ontologie de référence (domaine de la théorie des graphes), *graph* et *tree* soient deux termes associés à des concepts équivalents et qu'il existe un lien de subsumption entre le concept *tree* et le concept *structure*. D'après le tableau 5, et suivant la règle 7, la requête enrichie devient *publication&tree&(graph/computer)*. Supposons maintenant que l'ontologie ne contienne pas de concepts équivalents à *graph* mais seulement le lien de subsumption entre *tree* et *structure*. Ce sera alors la règle 8 qui s'appliquera pour donner *publication&(tree&structure)&computer*. Pour l'instant, on ne remonte que d'un niveau d'abstraction afin d'enrichir la requête, mais pour la suite de nos travaux, il est facilement imaginable de mettre en place un système interactif permettant à l'internaute de spécifier le type d'expansion souhaité.

Table 5. Règles d'expansion de requête

Requête initiale	Requête enrichie
$\neg \omega : O$	1) $\omega_1 \& (\omega_2 \dots \omega_n) \quad \forall i, 1 \leq i \leq n, \text{contraryOf}(\omega \ \omega_i)$ 2) $! \omega \& O$ s'il n'existe pas d'antonyme de w dans l'ontologie
$\omega : O$	3) $\omega \& (\omega_1 \dots \omega_n O)$ $\forall i, 1 \leq i \leq n, \text{equivalentClass}(\omega \ \omega_i) \vee \text{sameAs}(\omega \ \omega_i)$ 4) $\omega \& \omega_1 \& O$ si $\text{subClassOf}(\omega \ \omega_1) \vee \text{InstanceOf}(\omega \ \omega_1)$ 5) $\omega \& \omega_1 \& O$ si $\text{composedOf}(\omega \ \omega_1)$ 6) $\omega \& (!\omega_1) \& O$ si $\text{contraryOf}(\omega \ \omega_1)$
$\omega_1 \& \omega_2 : O$	7) $(\omega_1 \& \omega_2) \& (S_1 \dots S_n S_{n+1} \dots S_m O)$ $\forall i, 1 \leq i \leq n, \text{equivalentClass}(\omega_1 \ S_i) \vee \text{sameAs}(\omega_1 \ S_i)$ $\forall j, n+1 \leq j \leq m, \text{equivalentClass}(\omega_2 \ S_j) \vee \text{sameAs}(\omega_2 \ S_j)$ 8) $((\omega_1 \& h_1) \& (\omega_2 \& h_2)) ((\omega_1 \& h_1) \& \omega_2) (\omega_1 \& (\omega_2 \& h_2))) \& O$ si $(\text{subClassOf}(\omega_1 \ h_1) \wedge \text{subClassOf}(\omega_2 \ h_2)) \vee$ $(\text{InstanceOf}(\omega_1 \ h_1) \wedge \text{InstanceOf}(\omega_2 \ h_2))$

⁴ Ici le terme *computer* représente l'ontologie utilisée pour enrichir la requête

	9) $((\omega_1 \& h_1) \& (\omega_2 \& h_2)) \mid ((\omega_1 \& h_1) \& \omega_2) \mid (\omega_1 \& (\omega_2 \& h_2))) \& O$ <i>si composedOf(ω_1 h_1) \wedge composedOf(ω_2 h_2)</i>
	10) pas d'enrichissement si w_1 et w_2 sont contraires $\omega_1 \& \omega_2 \& (!a_1 \mid !a_2)$ avec <i>contraryOf(ω_1 a_1) \wedge contraryOf(ω_2 a_2)</i>
$\omega_1 \mid \omega_2 : O$	11) $(\omega_1 \mid \omega_2) \& (S_1 \mid \dots \mid S_n \mid S_{n+1} \mid \dots \mid S_m \mid O)$ $\forall i, 1 \leq i \leq n, \text{equivalentClass}(\omega_1 \ S_i) \vee \text{sameAs}(\omega_1 \ S_i)$ $\forall j, n+1 \leq j \leq m, \text{equivalentClass}(\omega_2 \ S_j) \vee \text{sameAs}(\omega_2 \ S_j)$
	12) $((\omega_1 \& h_1) \mid (\omega_2 \& h_2)) \& O$ <i>si (subClassOf(ω_1 h_1) \vee subClassOf(ω_2 h_2)) \vee</i> <i>(InstanceOf(ω_1 h_1) \wedge InstanceOf(ω_2 h_2))</i>
	13) $((\omega_1 \& h_1) \mid (\omega_2 \& h_2)) \& O$ <i>si composedOf(ω_1 h_1) \vee composedOf(ω_2 h_2)</i>
	14) $(\omega_1 \mid \omega_2) \& O$ <i>si contraryOf(ω_1 ω_2)</i> $(\omega_1 \mid \omega_2) \& (!a_1 \mid !a_2)$ avec <i>contraryOf(ω_1 a_1) \wedge contraryOf(ω_2 a_2)</i>
$\omega_1 \# \omega_2 : O$	15) $(\omega_1 \& (S_1 \mid \dots \mid S_n \mid O)) \# (\omega_2 \& (S_{n+1} \mid \dots \mid S_m \mid O))$ $\forall i, 1 \leq i \leq n, \text{equivalentClass}(\omega_1 \ S_i) \vee \text{sameAs}(\omega_1 \ S_i)$ $\forall j, n+1 \leq j \leq m, \text{equivalentClass}(\omega_2 \ S_j) \vee \text{sameAs}(\omega_2 \ S_j)$
	16) $(\omega_1 \& h_1 \& O) \# (\omega_2 \& h_2 \& O)$ <i>si (subClassOf(ω_1 h_1) \wedge subClassOf(ω_2 h_2)) \vee</i> <i>(InstanceOf(ω_1 h_1) \wedge InstanceOf(ω_2 h_2)) \vee</i> <i>(composedOf(ω_1 h_1) \wedge composedOf(ω_2 h_2))</i>
	17) $(\omega_1 \# \omega_2) \& O$ <i>si contraryOf(ω_1 ω_2)</i>

4 Résultats expérimentaux

Après avoir présenté, au cours des sections précédentes, les différents concepts de notre approche, nous allons présenter dans cette section une validation expérimentale des règles d'expansion de requêtes à travers le développement d'un prototype les mettant en œuvre. L'expérimentation a consisté à extraire les documents vérifiant une requête initiale formulée par un utilisateur grâce à Google et à les transformer en WPGraph et W³Graph selon une ontologie OWL (voir figure 1). La deuxième phase a consisté à enrichir la requête initiale par ajout des termes extraits de cette ontologie suivant les règles d'expansion explicitées dans la section 3.2. Finalement, la requête enrichie a été vérifiée sur les graphes afin d'en extraire les informations pertinentes. L'expérimentation porte sur une centaine de requêtes enrichies à l'aide de cinq ontologies différentes modélisant le même domaine mais en utilisant des relations ontologiques différentes afin de mettre en évidence l'apport de chaque règle. A titre indicatif, le tableau des résultats contient également des éléments de comparaisons de notre approche avec les moteurs de recherche usuels.

Table 5. Résultats expérimentaux

	Relations utilisées pour l'expansion de requête	Nombre moyen de pages retournées	Précision	Rappel
O ⁴	Equivalence	45	99%	1%
	Subsumption/Instance	38	99%	11%
	Composition	23	100%	46%
	Opposition	2	100%	85%
Google		65 000 000	40%	
Yahoo		27 000 000	33%	
Clusty		9 000 000	22%	

Les résultats de la table 5 montrent une très bonne précision de l'approche O⁴, bien meilleure que celle des outils de recherche les plus utilisés par les internautes. Ceci vient des règles d'expansion dont l'objectif est de cibler le domaine de recherche. Cependant, la complexité de la construction des WPGraphs et W³Graph entraîne des temps de réponse importants. Le fait de travailler sur un nombre fini de documents⁵ nous permet de calculer le rappel pour notre approche. On observera que le rappel croît si on privilégie l'enrichissement par des concepts généraux, des composés puis des concepts opposés. Ceci vient de fait que l'ajout de ces types de concepts permet de cibler des documents spécifiques et d'éviter de retourner les documents les plus généraux. Ainsi, on choisira en priorité l'enrichissement par des concepts équivalents si l'utilisateur souhaite des informations générales et on ajoutera des concepts liés par les autres types de relation si des documents plus spécifiques sont visés. Enfin, le classement des résultats est différent d'une approche à l'autre. Google, de part son algorithme de PageRank, parvient à classer les pages intéressantes parmi les meilleurs résultats alors que Yahoo range les informations intéressantes à la fin obligeant ainsi l'utilisateur à passer du temps pour atteindre ces pages Web. Clusty, quant à lui, utilise une méthode de classification des résultats offrant la possibilité à l'utilisateur de choisir un domaine auquel pourrait se rattacher la requête. Notre approche, grâce à sa plus grande précision, donne directement les résultats pertinents pour l'utilisateur.

5 Conclusion

Dans cet article, nous avons proposé une extension du langage de représentation d'ontologies OWL ainsi qu'une illustration de l'utilisation que l'on peut faire des concepts développés avec un tel langage à travers l'établissement de règles d'expansion de requêtes pour le Web dans le but d'optimiser la recherche de documents pertinents. L'approche proposée a montré, via les premiers résultats expérimentaux, une plus grande précision que celle obtenue par un certain nombre de moteurs. Cependant, le Web est un espace dynamique en constante évolution. Chaque jour, de nouvelles pages sont ajoutées, d'autres sont retirées, faisant évoluer, en vertu

⁵ On ne travaille que sur les 50 premiers résultats de Google

de la diversité des données modifiées, les domaines de recherche. L'internaute n'est pas toujours conscient de ces changements et par conséquent ne les intègre pas dans ses requêtes. L'idée de mettre à disposition des utilisateurs des ontologies dynamiques intégrant ces phénomènes d'évolution et proposant des relations ontologiques plus riches permettrait des recherches plus efficaces. Nous orientons nos travaux à venir dans cette direction.

Références

- BARBIER F., HENDERSON-SELLERS B., PARC-LACAYRELLE A. L. & BRUEL J.-M. (2003). Formalization of the whole-part relationship in the unified modeling language. *IEEE Transactions on Software Engineering*, **29**(5), 459-470.
- BERNERS-LEE T., HENDLER J. & LASSILA O. (2001). The semantic web. *Scientific American*, **284**(5), 34-43.
- CUI H., WEN J.-R., NIE J.-Y. & MA W.-Y. (2002). Probabilistic query expansion using query logs.: *Proceedings of the 11th international conference on World Wide Web*, p. 325-332, Honolulu, Hawaii, USA: ACM Press.
- EDMUNDSON H. P. (1967). Axiomatic characterization of synonymy and antonymy. In *Proceedings of the 1967 conference on Computational linguistics*, p. 1-11, Morristown, NJ, USA: Association for Computational Linguistics.
- FELLBAUM C. D. (1998). WordNet: An Electronic Lexical Database. Cambridge, MIT Press.
- GUELFI N. & PRUSKI C. (2006). On the use of ontologies for an optimal representation and exploration of the web. *Journal of Digital Information Management (JDIM)*, **4**(3).
- GUELFI N., PRUSKI C. & REYNAUD C. (2007). Towards the Adaptive Web using metadata evolution. In C. Calero, M. Á. Moraga & M. Piattini, Eds., *Handbook of research on Web information systems quality*. Idea Group Publishing.
- HIRST G. & ST-ONGE D. (1998). Lexical chains as representation of context for the detection and correction malapropisms. In C. Fellbaum, Ed., *WordNet: An electronic lexical database and some of its applications*, p. 305-332, Cambridge, MA: The MIT Press.
- JOHO H., COVERSON C., SANDERSON M. & BEAULIEU M. (2002). Hierarchical presentation of expansion terms. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, p. 645-649: ACM Press.
- LIN H.-C., WANG L.-H. & CHEN S.-M. (2006). Query expansion for document retrieval based on fuzzy rules and user relevance feedback techniques. *Expert Systems with Applications*, **31**(2), 397-405.
- MCGUINNESS D. & VAN HARMELEN F. (2004). OWL web ontology language overview. W3C Recommendation.
- NAVIGLI R. & VELARDI P. (2003). An analysis of ontology-based query expansion strategies. In *Proceeding of the Workshop on Adaptive Text Extraction and Mining*, Dubrovnik (Croatia).
- NECIB C. B. & FREYTAG J. C. (2004). Using ontologies for database query reformulation. In *ADBIS (Local Proceedings)*.
- OMG (2005). *UML 2.0 Superstructure Specification*. Technical Report formal/05-07-04, Object Management Group.